# Лабораторная работа №3

Исследование архитектуры и системы команд 16-разрядного процессора

## Цель работы

Исследовать систему команд, архитектуру и основные блоки процессора Intel 8086 и взаимодействие этих блоков процессора при выполнении команд разных типов. Приобрести практические навыки написания ассемблерных программ и отладки их в эмуляторе микропроцессора — экранным отладчиком типа emu8086.

## Программа работы

А. Изучить архитектуру МП 8086, состав регистров и работу процессора с использованием временных диаграмм.

Б. Исследовать процесс выполнения программы (т.е. проследить изменение содержимого регистров) сложения-вычитания, взятой из примера программ отладчика (code examples).

В. Исследовать процесс выполнения программы сложения элементов двух массивов (add\_two\_arrays), взятой из примера программ отладчика (code examples).

## Ход выполнения программы

А. Схема 16-разрядного микропроцессора представлена в Приложении Б, Рисунок Б.1.

Б. Для проведения исследований была взята программа, листинг которой представлен в Приложении Б. Ход выполнения программы:

1. В регистр AX было записано число 0005h, в BX – 000Ah.
2. В результате выполнения сложения в регистр BX было записано число 000Fh – результат сложения чисел из регистров AX и BX.
3. Из полученного на прошлом шаге результата было вычтено число 1. Новое значение в регистре – 000Eh.
4. Вывод полученного числа на экран и завершение работы программы.

В. Для проведения исследований была взята программа из Приложения Б. Ход выполнения программы:

1. После выполнения блока команд lea и последующей команды mov в регистры были записаны значения, представленные в Таблица 1.

Таблица – Результат выполнения группы команд

|  |  |  |
| --- | --- | --- |
| Регистр | Установленное значение | Пояснение |
| CS | 0700h | Установка значения сегмента текущего кода. |
| SI | 0102h | Индекс смещения для vec1 |
| BX | 0106h | Индекс смещения для vec2 |
| DI | 010Ah | Индекс смещения для vec3 |
| CX | 0004h | Счётчик цикла – количество элементов в массиве. В процессе выполнения программы будет уменьшаться на 1 до нуля |

1. Выполнение цикла sum, содержащего следующие действия:
   1. Запись в AL значения, находящегося по смещению SI (обрабатываемый элемент массива vec1).
   2. Прибавить к AL значения, находящегося по смещению BX (обрабатываемый элемент массива vec2).
   3. Записать в ячейку по смещению DI значение, содержащееся в регистре AL.
   4. Инкрементировать значения адресов обрабатываемых элементов массивов на 1.

Цикл будет выполняться до тех пор, пока значение CX не станет равно нулю.

1. Завершение выполнения программы.

# Вывод

ПРИЛОЖЕНИЕ А

Схемы

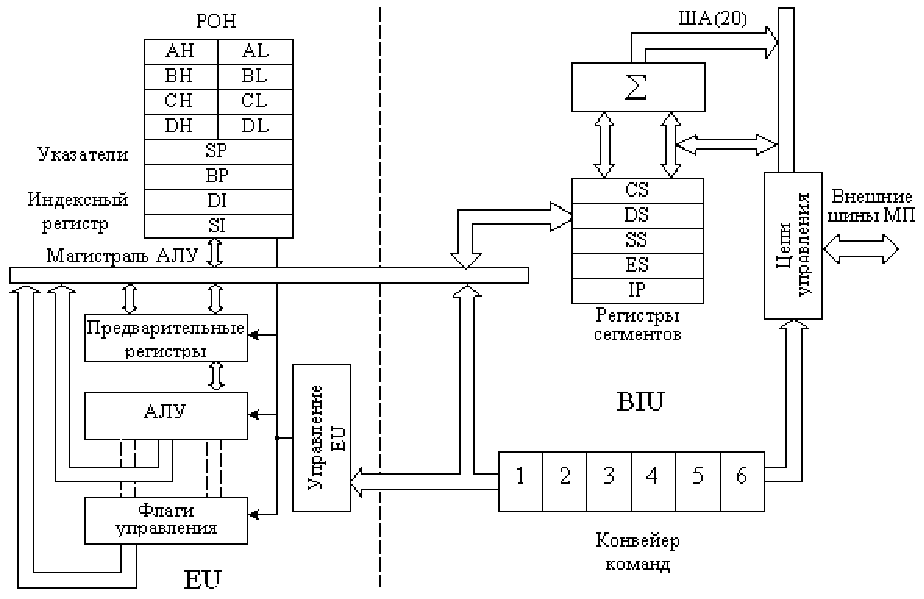


Рисунок Б. – Схема 16-разрядного микропроцессора K1810BM86

ПРИЛОЖЕНИЕ Б

Листинги программ

Б.1. Листинг программы проведения простых арифметических операций над двумя числами

name "add-sub"

org 100h

mov al, 5 ; bin=00000101b

mov bl, 10 ; hex=0ah or bin=00001010b

; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b

add bl, al

; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b

sub bl, 1

; print result in binary:

mov cx, 8

print: mov ah, 2 ; print function.

mov dl, '0'

test bl, 10000000b ; test first bit.

jz zero

mov dl, '1'

zero: int 21h

shl bl, 1

loop print

; print binary suffix:

mov dl, 'b'

int 21h

; wait for any key press:

mov ah, 0

int 16h

ret

Б.2. Листинг программы сложения элементов двух массивов

name "add-2"

; this example calculates the sum of a vector with

; another vector and saves result in third vector.

; you can see the result if you click the "vars" button.

; set elements for vec1, vec2 and vec3 to 4 and show as "signed".

org 100h

jmp start

vec1 db 1, 2, 5, 6

vec2 db 3, 5, 6, 1

vec3 db ?, ?, ?, ?

start:

lea si, vec1

lea bx, vec2

lea di, vec3

mov cx, 4

sum:

mov al, [si]

add al, [bx]

mov [di], al

inc si

inc bx

inc di

loop sum

ret